**졸업 프로젝트**

**요구사항 분석서 (2차)**

-아두이노와 웹서버를 이용한 스마트홈 오토메이션-

고지혜 201311193

김예찬 201311200

1. 개요

1.1 프로젝트 기획 배경

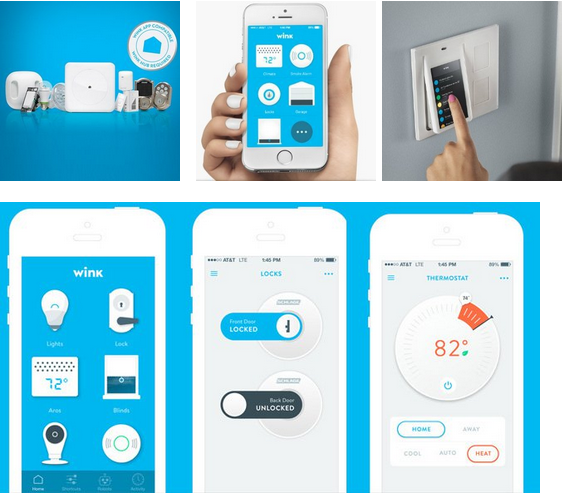
프로젝트의 주제에 대해서 교수님과 토론하는 중에 교수님의 연구실에 있는 아두이노를 발견하고, 이에 대해서 교수님과 다시 얘기를 해보면서 흥미 있는 주제라고 생각을 하게 되었다.

또한 현재 IT분야에서 스마트홈에 대한 주제는 미래 가망성이 높다고 평가되고 있어 프로젝트 의 주제로 적절하다고 생각하여 이번 프로젝트의 주제로 선정하였다.

1.2 기술 동향

최근 들어 미국, 중국 등의 해외 시장을 중심으로 스마트홈에 대한 소비자들의 관심이 높아지고 있다. 시장 조사업체 마켓앤마켓은 전 세계 스마트홈 시장이 2015년부터 연평균 17%씩 성장해 2020년에는 거의 600억 달러에 달할 것으로 내다보고 있다.

이렇게 많은 해외 시장에서 스마트홈에 대한 연구는 많이 진행되고 있다고 한다. 예를 들어, 미국의 스타트업 퀼키(Quirky)가 스마트홈 시장 진출을 위해 윙크(Wink)를 설립했는데, 퀼키는 단돈 50달러짜리 윙크 허브를 통해 다양한 브랜드의 스마트홈 제품을 연동해 제어할 수 있는 스마트홈 플랫폼을 만들었다. 이 윙크 허브를 통해 조명, 전원관리, IP 카메라, 화재감지, 도어락, 가전, 블라인드 등을 관리한다고 한다.



국내 기업에서도 스마트홈에 대한 주제로 연구가 활발하고, 대표적인 기업으로는 LG를 들을 수 있다. LG에서는 스마트싱큐(SmartThinQ) 라는 메인 센서를 가전제품에 설치하여 스마트폰과 태블릿으로 제어를 할 수 있고, 그 가전제품의 상태를 확인할 수도 있다고 한다. 또한 그 센서에는 진동을 포함한 4개의 센서를 가지고 있는데, 이를 통해 냉장고, 세탁기등의 진동, 온도를 확인하여 식품의 상태, 세탁의 완료 여부 등 더욱 세밀한 상태를 알 수 있다고 한다.



1.3 프로젝트 주요 기능 및 특징

1) 전자제품/조명 키고 끄기, 커튼 열고 닫기

안드로이드 앱에서 LTE및 Wifi를 이용하여 서버로 정보를 보내면 그 정보를 구분(파싱)하여 중앙아두이노에서 해당 되는 물체의 아두이노에 신호를 보내어 제어한다. 집과 멀리 떨어진 외부에서도 와이파이나 데이터 통신을 이용하여 서버에 접근할 수 있으므로 아두이노로 명령을 보내는 것이 가능해진다.

2) GPS를 이용해 사용자가 집 근처로 들어오면 전자제품을 키기

사용자가 외출한 후 집에 돌아올 때 에어컨, 보일러, 컴퓨터(부팅이 오래 걸리는) 등 사용자가 집에 도착하기 전에 미리 틀어놓으면 좋은 기기들은 사용자의 GPS의 위도 경도와 중앙 아두이노 의 GPS의 위도 경도의 거리를 계산하여 적정 값 이하일 때 에어컨을 틀도록 신호를 보낸다.

외출 시에 계속해서 서버에 GPS의 위도 경도를 보내게 되면 데이터 통신료가 많이 들게 되므로 안드로이드 앱에서 시간당 위, 경도의 변화가 적절한 값이 되는 주기를 계산하여 그 주기에 맞춰 데이터 통신을 하도록 한다.

3) 전자제품/조명의 상태 알리기

집 안의 전자제품과 조명의 On/Off상태를 안드로이드 앱에서 보여준다. On/Off상태는 변경이 있을 때 마다 서버로 보내어 안드로이드 앱으로 보낸다.

4) 행동 패턴 분석

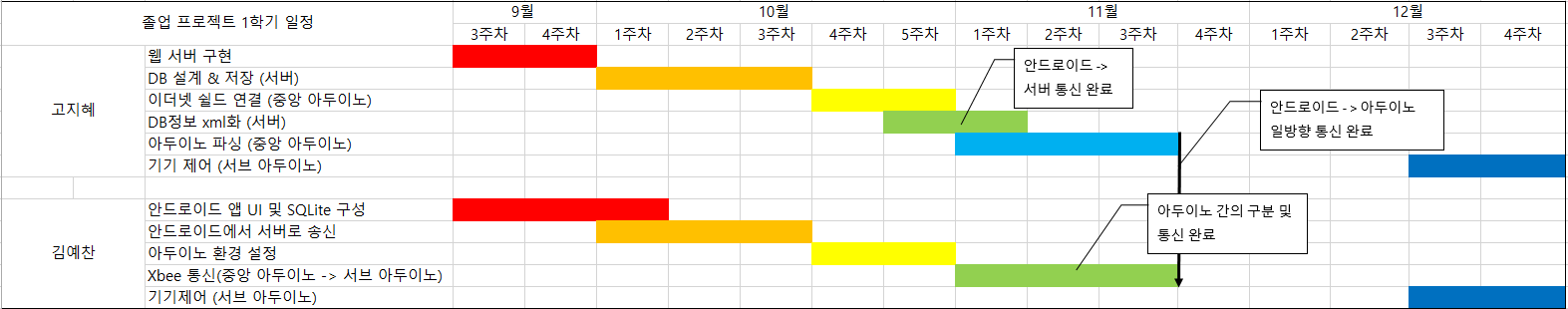
사용자가 보내온 명령들의 주기, 빈도 등을 바탕으로 행동 패턴을 분석하여 아두이노가 스스로 사물들을 제어한다.

1.4 조원 구성 및 역할 분담

|  |  |
| --- | --- |
| -고지혜-  서버 사이드에서 명령 가공 및 DB저장  중앙 아두이노에서 서버로부터 명령 수신  중앙 아두이노에서 서버로 사물들의 상태 송신 | -김예찬-  안드로이드 앱에서 서버로 명령 송신  중앙 아두이노에서 서브 아두이노로 정보 송신  중앙에서 안드로이드 앱으로 사물 상태 송신 |
| -공통  웹 서버 구축  사물 제어  GPS 알고리즘 | |

1.5 일정

1학기 일정



2학기 일정



2 기능적 요구사항

2.1 Top Level Use Case Diagram



1) 안드로이드 기기 등록

새로운 user는 자신의 기기로 서비스를 받기 위해 안드로이드 기기를 등록할 수 있다. 등록되지 않은 기기는 제어가 불가능하게 함으로서 보안을 강화한다.

2) 사물 상태 확인

아두이노가 설치된 사물의 현재 상태를 사용자의 안드로이드 기기에서 확인할 수 있다. 사용자는 상태를 확인하고 싶은 사물을 선택할 수 있다.

3) 사물 제어 명령

안드로이드 기기에서 원거리에 있는 사물들을 제어할 수 있다. 사용자는 제어할 사물을 선택할 수 있고, 해당 사물에 대한 제어 내용을 선택할 수 있다. 제어할 내용은 해당 사물을 키거나 끄는 경우(에어컨, 조명의 경우), 열거나 닫는 경우(커튼)를 선택할 수 있다.

4) 자동 제어 설정

사용자는 자동 제어를 켜놓을지(On), 꺼놓을지(Off)를 정할 수 있다. 자동 제어는 사용자의 위치(안드로이드 기기)와 중앙 아두이노의 위치를 계산해서 사용자가 일정 거리 이내에 들어왔을 때 특정한 제어하거나, 사용자의 제어 패턴(시간, 상태)을 토대로 자동으로 제어한다.

2.2 Use case document

|  |  |
| --- | --- |
| *Use case name* | 제어 사물 상태 확인 |
| *Participating actors* | 등록된 사용자 (Old User)  보안 시스템  사물 체크 시스템 |
| *Flow of events* | 1. 사용자는 앱을 통해 ‘상태확인’ 기능을 메뉴를 통해 실행한다.  2. 보안 시스템은 사용자가 등록된 사용자인지 확인을한다. 등록된 사용자라면 사물체크 시스템은 사용자에게 사물의 현재 상태정보를 보낸다.  3. 사용자는 앱을 통해 사물의 현재 상태를 확인한다. |
| *Entry condition* | • 사용자의 안드로이드 기기는 시스템에 등록되어 있어야 함. |
| *Exit condition* | • 사용자가 시스템으로부터 원하는 기기의 상태를 받음. |
| *Special Requirements* | • 사용자는 제어 액티비티에서 제어한 후 변경된 상태에 대해서 5초 내에 받아볼 수 있어야 함. |

|  |  |
| --- | --- |
| *Use case name* | 사용자 자동 제어 |
| *Participating actors* | 등록된 사용자 (Old User)  패턴 분석 시스템  사물 제어 시스템 |
| *Flow of events* | 1. 패턴 분석 시스템은 사용자의 제어 기록과 상태를 바탕으로 패턴을 추출하고, 그 패턴에 대한 정보를 사용자의 앱에 보낸다.  2. 사용자는 자신의 ‘자동제어’ 기능 화면에서 자신의 패턴을 확인하고, On / Off 를 설정한다.  3. 패턴 분석 시스템은 사용자의 패턴에 따른 자동제어상태가 On인지를 확인한다. On이라면 사용자의 현재상태와 패턴에 저장된 상태를 비교하고, 그 차이가 오차 범위이내라면 사물 제어 시스템에 제어 명령을 보낸다.  4. 사물제어 시스템은 ‘사물 제어’ 기능을 실행한다. |
| *Entry condition* | • 사용자의 안드로이드 기기는 시스템에 등록되어 있어야 함.  • 서버의 패턴 알고리즘으로부터 사용자의 패턴이 추출되어야 함. |
| *Exit condition* | • 기기가 자동으로 제어가 됨. |
| *Special Requirements* | • 사용자 패턴에 대한 서비스는 7일을 주기로 갱신되어야 한다. |

|  |  |
| --- | --- |
| *Use case name* | 기기 등록 |
| *Participating actors* | 새로운 사용자 (New User)  보안 시스템 |
| *Flow of Events* | 1. 안드로이드 앱을 처음 실행할 때 기기 등록 화면이 뜨게 된다.  2. 시스템은 등록 form을 제공한다. Form에는 사용자의 이름과 출근시간, 퇴근시간 등의 생활 패턴에 대한 입력 칸이 있다.  3. 사용자가 form을 작성하고 submit한다.  4. 보안 시스템은 관리자 페이지를 제공한다.  5. 해당 페이지에서 관리자는 전송된 form 데이터를 검토하여 해당 기기를 등록할지, 등록하지 않을지 선택한다.  6. 등록하기를 선택하면 해당 기기와 사용자의 정보가 데이터베이스에 들어가게 된다. |
| *Entry condition* | • 안드로이드 앱은 데이터 통신이나 wifi등을 이용하여 인터넷에 연결할 수 있는 상태여야 한다.  • 웹 서버가 동작 중이어야 한다. |
| *Exit condition* | • 데이터베이스에 기기가 등록됨 |

|  |  |
| --- | --- |
| *Use case name* | 사물 제어 |
| *Participating actors* | 등록된 사용자 (Old User)  보안 시스템  사물 제어 시스템 |
| *Flow of Events* | 1. 사용자가 안드로이드 앱 메인 화면에서 사물 제어 메뉴를 누른다.  2. 시스템 앱은 사물 제어 화면에서 어떤 사물을 제어할지 세가지중 한가지를 선택하도록 메뉴 화면을 제공한다.  3. 사용자가 사물 제어 화면에서 어떤 사물을 제어할지 세가지중 한가지를 선택하여 누른다.  4. 시스템 앱은 선택한 사물을 어떻게 제어할 것인지 한가지를 선택할 수 있도록 선택지를 제공한다. 선택지에는 에어컨과 조명의 경우에는 On/Off가 있고, 창문의 경우에는 Open/Close가 있다.  5. 사용자가 선택한 사물을 어떻게 제어할 것인지 한가지를 선택한다.  6. 시스템이 제어 명령을 사물로 전송한다. |
| *Entry condition* | • 안드로이드 앱은 데이터 통신이나 wifi등을 이용하여 인터넷에 연결할 수 있는 상태여야 한다.  • 웹 서버가 동작 중이어야 한다.  • 중앙 아두이노와 서브 아두이노 모두 전원이 켜진 상태여야 한다. |
| *Exit condition* | • 사물이 해당 명령에 따라서 제어된다. |
| *Special Requirements* | • 사물 제어는 명령을 보낸 시점으로부터 5초 이내에 수행되어야 한다. |

3 비기능적 요구사항

3.1 사용편리성

3.1.1 UI

* 시작화면에서는 사물 제어, 사물 상태보기, 사용자 설정 메뉴가 배치되어야 한다.
* 사물 제어 화면(Activity)에는 제어 가능한 사물의 메뉴가 배치되어야 한다.
* 각 사물의 메뉴에는 사물을 제어할 수 있는 버튼이 배치되어야 한다.
* 사물 상태보기 화면(Activity)에는 모든 제어 가능한 사물의 상태가 수직적으로 배치되어야 한다.

3.1.2 오류

* 서비스 중에 일어나는 오류사항들에 대해서는 다이얼로그 형태로 보여주어야 한다.

3.1.3 지원 언어

* 모든 인터페이스는 영어로 지원되어야 한다.

3.2 신뢰성

* 안드로이드 기기가 네트워크에 접속할 수 있는 환경이고 웹 서버가 정상적으로 동작 중이라면, 사물 제어 기능은 100%의 확률로 수행되어야 한다.
* 안드로이드 기기가 네트워크에 접속할 수 있는 환경이고 웹 서버가 정상적으로 동작 중이라면, 사물 상태보기 기능은 100%의 확률로 실시간의 사물 상태를 반영해야 한다.

3.3 성능

* 사물 제어 시간은 5초 이내여야 한다.
* 사물 상태를 불러오는 시간은 3초 이내여야 한다.
* 사용자 패턴에 대한 서비스는 7일을 주기로 갱신되어야 한다.

3.4 이식성 요구 사항

* 웹 서버는 Python3가 지원되는 환경이어야 한다.
* 서비스를 받는 안드로이드 기기는 안드로이드 버전 4.1(Jelly bean) 이상이어야 한다.

3.5 법적 요구 사항

* 사용자의 정보가 저장되어 있는 데이터베이스는 외부로부터 접근할 수 없어야 한다.